

MUSIC STRUCTURE DETECTION APPARATUS AND METHOD

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to an apparatus and a method for detecting the structure of a music piece in accordance with data representing chronological changes in chords in the music piece.

2. Description of the Related Background Art

In popular music in general, phrases are expressed as introduction, melody A, melody B and release, and melody A, melody B, and release parts are repeated a number of times, as a refrain. The release phrase for a so-called heightened part of a music piece in particular is more often selectively used than the other parts when the music is included in a music program or a commercial message aired on radio or TV broadcast. Generally, each of the phrases is determined by actually listening to the sound of the music piece before broadcasting.

If how the phrases including the release part of a music piece is repeated, in other words, the overall structure of the music piece can be understood, not only the release part but also the other repeating phrases can easily be selectively played. However, since there has been no such apparatus that automatically detects the overall structure of music pieces, the user has no choice but actually listen to the music to determine phrases as mentioned above.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to provide an apparatus and a method allowing the structure of a music piece including repeating parts to be appropriately detected with a simple structure.

A music structure detection apparatus according to the present invention which detects a structure of a music piece in accordance with chord progression music data representing chronological changes in chords in the music piece, comprising: a partial music data producing device which produces partial music data pieces each including a predetermined number of consecutive chords starting from a position of each chord in the chord progression music data; a comparator which compares each of the partial music data pieces with the chord progression music data from each of the starting chord positions in the chord progression music data, on the basis of an amount of change in a root of a chord in each chord transition and an attribute of the chord after the transition, thereby calculating degrees of similarity for each of the partial music data pieces; a chord position detector which detects a position of a chord in the chord progression music data where the calculated similarity degree indicates a peak value higher than a predetermined value for each of the partial music data pieces; and an output device which calculates the number of times that the calculated similarity degree indicates a peak value higher than the predetermined value for all the

partial music data pieces for each chord position in the chord progression music data, thereby producing a detection output representing the structure of the music piece in accordance with the calculated number of times for each chord position.

A method according to the present invention which detects a structure of a music piece in accordance with chord progression music data representing chronological changes in chords in the music piece, the method comprising the steps of: producing partial music data pieces each including a predetermined number of consecutive chords starting from a position of each chord in the chord progression music data; comparing each of the partial music data pieces with the chord progression music data from each of the starting chord positions in the chord progression music data, on the basis of an amount of change in a root of a chord in each chord transition and an attribute of the chord after the transition, thereby calculating degrees of similarity for each of the partial music data pieces; detecting a position of a chord in the chord progression music data where the calculated similarity degree indicates a peak value higher than a predetermined value for each of the partial music data pieces; and calculating the number of times that the calculated similarity degree indicates a peak value higher than the predetermined value for all the partial music data pieces for each chord position in the chord progression music data, thereby producing a detection

output representing the structure of the music piece in accordance with the calculated number of times for each chord position.

A computer program product according to the present invention comprising a program for detecting a structure of a music piece, the detecting comprising the steps of: producing partial music data pieces each including a predetermined number of consecutive chords starting from a position of each chord in the chord progression music data; comparing each of the partial music data pieces with and the chord progression music data from each of the starting chord positions in the chord progression music data, on the basis of an amount of change in a root of a chord in each chord transition and an attribute of the chord after the transition, thereby calculating degrees of similarity for each of the partial music data pieces; detecting a position of a chord in the chord progression music data where the calculated similarity degree indicates a peak value higher than a predetermined value for each of the partial music data pieces; and calculating the number of times that the calculated similarity degree indicates a peak value higher than the predetermined value for all the partial music data pieces for each chord position in the chord progression music data, thereby producing a detection output representing the structure of the music piece in accordance with the calculated number of times for each chord position.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of the configuration of a music processing system to which the invention is applied;

Fig. 2 is a flow chart showing the operation of frequency error detection;

Fig. 3 is a table of ratios of the frequencies of twelve tones and tone A one octave higher with reference to the lower tone A as 1.0;

Fig. 4 is a flow chart showing a main process in chord analysis operation;

Fig. 5 is a graph showing one example of the intensity levels of tone components in band data;

Fig. 6 is a graph showing another example of the intensity levels of tone components in band data;

Fig. 7 shows how a chord with four tones is transformed into a chord with three tones;

Fig. 8 shows a recording format into a temporary memory;

Figs. 9A to 9C show method for expressing fundamental notes of chords, their attributes, and a chord candidate;

Fig. 10 is a flow chart showing a post-process in chord analysis operation;

Fig. 11 shows chronological changes in first and second chord candidates before a smoothing process;

Fig. 12 shows chronological changes in first and second chord candidates after the smoothing process;

Fig. 13 shows chronological changes in first and second chord candidates after an exchanging process;

Figs. 14A to 14D show how chord progression music data is produced and its format;

Fig. 15 is a flow chart showing music structure detection operation;

Fig. 16 is a chart showing a chord differential value in a chord transition and the attribute after the transition;

Fig. 17 shows the relation between chord progression music data including temporary data and partial music data;

Figs. 18A to 18C show the relation between the C-th chord progression music data and chord progression music data for a search object, changes of a correlation coefficient $COR(t)$, time widths for which chords are maintained, jump processes, and a related key process;

Figs. 19A to 19F show changes of the correlation coefficient $COR(c, t)$ corresponding to a phrase included in partial music data and a line of phrases included in chord progression music data;

Fig. 20 shows peak numbers $PK(t)$ for a music piece having the phrase line in Figs. 19A to 19F and a position $COR_PEAK(c, t)$ where a peak value is obtained;

Fig. 21 shows the format of music structure data;

Fig. 22 shows an example of display at a display device; and

Fig. 23 is a block diagram of the configuration of a music processing system as another embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Hereinafter, embodiments of the present invention will be described in detail with reference to the drawings.

Fig. 1 shows a music processing system to which the present invention is applied. The music processing system includes a music input device 1, an input operation device 2, a chord analysis device 3, data storing devices 4 and 5, a temporary memory 6, a chord progression comparison device 7, a repeating structure detection device 8, a display device 9, a music reproducing device 10, a digital-analog converter 11, and a speaker 12.

The music input device 1 is, for example, a CD player connected with the chord analysis device 3 and the data storing device 5 to reproduce a digitized audio signal (such as PCM data). The input operation device 2 is a device for a user to operate for inputting data or commands to the system. The output of the input operation device 2 is connected with the chord analysis device 3, the chord progression comparison device 7, the repeating structure detection device 8, and the music reproducing device 10. The data storing device 4 stores the music data (PCM data) supplied from the music input device 1 as files.

The chord analysis device 3 analyzes chords of the supplied music data by chord analysis operation that will be described. The chords of the music data analyzed by the chord analysis device 3 are temporarily stored as first and second chord candidates in the temporary memory 6. The data

storing device 5 stores chord progression music data analyzed by the chord analysis device 3 as a file for each music piece.

The chord progression comparison device 7 compares the chord progression music data stored in the data storing device 5 with a partial music data piece that constitutes a part of the chord progression music data to calculate degrees of similarity. The repeating structure detection device 8 detects a repeating part in the music piece using a result of the comparison by the chord progression music comparison device 7.

The display device 9 displays the structure of the music piece including its repeating part detected by the repeating structure detection device 8.

The music reproducing device 10 reads out the music data for the repeating part detected by the repeating structure detection device 8 from the data storing device 4 and reproduces the data for sequential output as a digital audio signal. The digital-analog converter 11 converts the digital audio signal reproduced by the music reproducing device 10 into an analog audio signal for supply to the speaker 12.

The chord analysis device 3, the chord progression comparison device 7, the repeating structure detection device 8, and the music reproducing device 10 operate in response to each command from the input operation device 2.

Now, the operation of the music processing system

having the structure will be described.

Here, assume that a digital audio signal representing music sound is supplied from the music input device 1 to the chord analysis device 3.

The chord analysis operation includes a pre-process, a main process, and a post-process. The chord analysis device 3 carries out frequency error detection operation as the pre-process.

In the frequency error detection operation, as shown in Fig. 2, a time variable T and a band data F(N) each are initialized to zero, and a variable N is initialized, for example, to the range from -3 to 3 (step S1). An input digital signal is subjected to frequency conversion by Fourier transform at intervals of 0.2 seconds, and as a result of the frequency conversion, frequency information f(T) is obtained (step S2).

The present information f(T), previous information f(T-1), and information f(T-2) obtained two times before are used to carry out a moving average process (step S3). In the moving average process, frequency information obtained in two operations in the past are used on the assumption that a chord hardly changes within 0.6 seconds. The moving average process is carried out by the following expression:

$$f(T) = (f(T) + f(T-1)/2.0 + f(T-2)/3.0) / 3.0 \quad \dots (1)$$

After step S3, the variable N is set to -3 (step S4), and it is determined whether or not the variable N is smaller than 4 (step S5). If $N < 4$, frequency components

$f_1(T)$ to $f_5(T)$ are extracted from the frequency information $f(T)$ after the moving average process (steps S6 to S10). The frequency components $f_1(T)$ to $f_5(T)$ are in tempered twelve tone scales for five octaves based on $110.0+2 \times N$ Hz as the fundamental frequency. The twelve tones are A, A#, B, C, C#, D, D#, E, F, F#, G, and G#. Fig. 3 shows frequency ratios of the twelve tones and tone A one octave higher with reference to the lower tone A as 1.0. Tone A is at $110.0+2 \times N$ Hz for $f_1(T)$ in step S6, at $2 \times (110.0+2 \times N)$ Hz for $f_2(T)$ in step S7, at $4 \times (110.0+2 \times N)$ Hz for $f_3(T)$ in step S8, at $8 \times (110.0+2 \times N)$ Hz for $f_4(T)$ in step S9, and at $16 \times (110.0+2 \times N)$ Hz for $f_5(T)$ in step 10.

After steps S6 to S10, the frequency components $f_1(T)$ to $f_5(T)$ are converted into band data $F'(T)$ for one octave (step S11). The band data $F'(T)$ is expressed as follows:

$$F'(T) = f_1(T) \times 5 + f_2(T) \times 4 + f_3(T) \times 3 + f_4(T) \times 2 + f_5(T) \quad \dots (2)$$

More specifically, the frequency components $f_1(T)$ to $f_5(T)$ are respectively weighted and then added to each other. The band data $F'(T)$ for one octave is added to the band data $F(N)$ (step S12). Then, one is added to the variable N (step S13), and step S5 is again carried out.

The operations in steps S6 to S13 are repeated as long as $N < 4$ stands in step S5, in other words, as long as N is in the range from -3 to +3. Consequently, the tone component $F(N)$ is a frequency component for one octave including tone interval errors in the range from -3 to +3.

If $N \geq 4$ in step S5, it is determined whether or not

the variable T is smaller than a predetermined value M (step S14). If $T < M$, one is added to the variable T (step S15), and step S2 is again carried out. Band data $F(N)$ for each variable N for frequency information $f(T)$ by M frequency conversion operations is produced.

If $T \geq M$ in step S14, in the band data $F(N)$ for one octave for each variable N , $F(N)$ having the frequency components whose total is maximum is detected, and N in the detected $F(N)$ is set as an error value X (step S16).

In the case of existing a certain difference between the tone intervals of an entire music sound such as a performance sound by an orchestra, the tone intervals can be compensated by obtaining the error value X by the pre-process, and the following main process for analyzing chords can be carried out accordingly.

Once the operation of detecting frequency errors in the pre-process ends, the main process for analyzing chords is carried out. Note that if the error value X is available in advance or the error is insignificant enough to be ignored, the pre-process can be omitted. In the main process, chord analysis is carried out from start to finish for a music piece, and therefore an input digital signal is supplied to the chord analysis device 3 from the starting part of the music piece.

As shown in Fig. 4, in the main process, frequency conversion by Fourier transform is carried out to the input digital signal at intervals of 0.2 seconds, and frequency

information $f(T)$ is obtained (step S21). This step S21 corresponds to a frequency converter (FOR EP: conversion means). The present information $f(T)$, the previous information $f(T-1)$, and the information $f(T-2)$ obtained two times before are used to carry out moving average process (step S22). The steps S21 and S22 are carried out in the same manner as steps S2 and S3 as described above.

After step S22, frequency components $f_1(T)$ to $f_5(T)$ are extracted from frequency information $f(T)$ after the moving average process (steps S23 to S27). Similarly to the above described steps S6 to S10, the frequency components $f_1(T)$ to $f_5(T)$ are in the tempered twelve tone scales for five octaves based on $110.0+2 \times N$ Hz as the fundamental frequency. The twelve tones are A, A#, B, C, C#, D, D#, E, F, F#, G, and G#. Tone A is at $110.0+2 \times N$ Hz for $f_1(T)$ in step S23, at $2 \times (110.0+2 \times N)$ Hz for $f_2(T)$ in step S24, at $4 \times (110.0+2 \times N)$ Hz for $f_3(T)$ in step S25, at $8 \times (110.0+2 \times N)$ Hz for $f_4(T)$ in step S26, and at $16 \times (110.0+2 \times N)$ Hz for $f_5(T)$ in step 27. Here, N is X set in step S16.

After steps S23 to S27, the frequency components $f_1(T)$ to $f_5(T)$ are converted into band data $F'(T)$ for one octave (step S28). The operation in step S28 is carried out using the expression (2) in the same manner as step S11 described above. The band data $F'(T)$ includes tone components. These steps S23 to S28 correspond to a component extractor (FOR EP: extraction means).

After step S28, the six tones having the largest

intensity levels among the tone components in the band data $F'(T)$ are selected as candidates (step S29), and two chords M1 and M2 of the six candidates are produced (step S30). One of the six candidate tones is used as a root to produce a chord with three tones. More specifically, ${}_6C_3$ chords are considered. The levels of three tones forming each chord are added. The chord whose addition result value is the largest is set as the first chord candidate M1, and the chord having the second largest addition result is set as the second chord candidate M2.

When the tone components of the band data $F'(T)$ show the intensity levels for twelve tones as shown in Fig. 5, six tones, A, E, C, G, B, and D are selected in step S29. Triads each having three tones from these six tones A, E, C, G, B, and D are chord Am (of tones A, C, and E), chord C (of tones C, E, and G), chord Em (of tones E, B, and G), chord G (of tones G, B, and D),.... The total intensity levels of chord Am (A, C, E), chord C (C, E, G), chord Em (E, B, G), and chord G (G, B, D) are 12, 9, 7, and 4, respectively. Consequently, in step S30, chord Am whose total intensity level is the largest, i.e., 12 is set as the first chord candidate M1. Chord C whose total intensity level is the second largest, i.e., 9 is set as the second chord candidate M2.

When the tone components in the band data $F'(T)$ show the intensity levels for the twelve tones as shown in Fig. 6, six tones C, G, A, E, B, and D are selected in step S29.

Triads produced from three tones selected from these six tones C, G, A, E, B, and D are chord C (of tones C, E, and G), chord Am (of A, C, and E), chord Em (of E, B, and G), chord G (of G, B, and D), The total intensity levels of chord C (C, E, G), chord Am (A, C, E), chord Em (E, B, G), and chord G (G, B, D) are 11, 10, 7, and 6, respectively. Consequently, chord C whose total intensity level is the largest, i.e., 11 in step S30 is set as the first chord candidate M1. Chord Am whose total intensity level is the second largest, i.e., 10 is set as the second chord candidate M2.

The number of tones forming a chord does not have to be three, and there is, for example, a chord with four tones such as 7th and diminished 7th. Chords with four tones are divided into two or more chords each having three tones as shown in Fig. 7. Therefore, similarly to the above chords of three tones, two chord candidates can be set for these chords of four tones in accordance with the intensity levels of the tone components in the band data $F'(T)$.

After step S30, it is determined whether or not there are chords as many as the number set in step S30 (step S31). If the difference in the intensity level is not large enough to select at least three tones in step 30, no chord candidate is set. This is why step S31 is carried out. If the number of chord candidates > 0 , it is then determined whether the number of chord candidates is greater than one (step S32).

If it is determined in step S31 that the number of chord candidates = 0, the chord candidates M1 and M2 set in the previous main process at T-1 (about 0.2 seconds before) are set as the present chord candidates M1 and M2 (step S33). If the number of chord candidates = 1 in step S32, it means that only the first candidate M1 has been set in the present step S30, and therefore the second chord candidate M2 is set as the same chord as the first chord candidate M1 (step S34). These steps S29 to S34 correspond to a chord candidate detector (FOR EP: detection means).

If it is determined that the number of chord candidates > 1 in step S32, it means that both the first and second chord candidates M1 and M2 are set in the present step S30, and therefore, time, and the first and second chord candidates M1 and M2 are stored in the temporary memory 6 (step S35). The time and first and second chord candidates M1 and M2 are stored as a set in the temporary memory 6 as shown in Fig. 8. The time is the number of how many times the main process is carried out and represented by T incremented for each 0.2 seconds. The first and second chord candidates M1 and M2 are stored in the order of T.

More specifically, a combination of a fundamental tone (root) and its attribute is used in order to store each chord candidate on a 1-byte basis in the temporary memory 6 as shown in Fig. 8. The fundamental tone indicates one of the tempered twelve tones, and the attribute indicates a type of chord such as major {4, 3}, minor {3, 4}, 7th

candidate {4, 6}, and diminished 7th (dim7) candidate {3, 3}. The numbers in the braces { } represent the difference among three tones when a semitone is 1. A typical candidate for 7th is {4, 3, 3}, and a typical diminished 7th (dim7) candidate is {3, 3, 3}, but the above expression is employed in order to express them with three tones.

As shown in Fig. 9A, the 12 fundamental tones are each expressed on a 16-bit basis (in hexadecimal notation). As shown in Fig. 9B, each attribute, which indicates a chord type, is represented on a 16-bit basis (in hexadecimal notation). The lower order four bits of a fundamental tone and the lower order four bits of its attribute are combined in that order, and used as a chord candidate in the form of eight bits (one byte) as shown in Fig. 9C.

Step S35 is also carried out immediately after step S33 or S34 is carried out.

After step S35 is carried out, it is determined whether the music has ended (step S36). If, for example, there is no longer an input analog audio signal, or if there is an input operation indicating the end of the music from the input operation device 2, it is determined that the music has ended. The main process ends accordingly.

Until the end of the music is determined, one is added to the variable T (step S37), and step S21 is carried out again. Step S21 is carried out at intervals of 0.2 seconds, in other words, the process is carried out again after 0.2 seconds from the previous execution of the process.

In the post-process, as shown in Fig. 10, all the first and second chord candidates $M1(0)$ to $M1(R)$ and $M2(0)$ to $M2(R)$ are read out from the temporary memory 6 (step S41). Zero represents the starting point and the first and second chord candidates at the starting point are $M1(0)$ and $M2(0)$. The letter R represents the ending point and the first and second chord candidates at the ending point are $M1(R)$ and $M2(R)$. These first chord candidates $M1(0)$ to $M1(R)$ and the second chord candidates $M2(0)$ to $M2(R)$ thus read out are subjected to smoothing (step S42). The smoothing is carried out to cancel errors caused by noise included in the chord candidates when the candidates are detected at the intervals of 0.2 seconds regardless of transition points of the chords. As a specific method of smoothing, it is determined whether or not a relation represented by $M1(t-1) \neq M1(t)$ and $M1(t) \neq M1(t+1)$ stand for three consecutive first chord candidates $M1(t-1)$, $M1(t)$ and $M1(t+1)$. If the relation is established, $M1(t)$ is equalized to $M1(t+1)$. The determination process is carried out for each of the first chord candidates. Smoothing is carried out to the second chord candidates in the same manner. Note that rather than equalizing $M1(t)$ to $M1(t+1)$, $M1(t+1)$ may be equalized to $M1(t)$.

After the smoothing, the first and second chord candidates are exchanged (step S43). There is little possibility that a chord changes in a period as short as 0.6 seconds. However, the frequency characteristic of the

signal input stage and noise at the time of signal input can cause the frequency of each tone component in the band data $F'(T)$ to fluctuate, so that the first and second chord candidates can be exchanged within 0.6 seconds. Step S43 is carried out as a remedy for the possibility. As a specific method of exchanging the first and second chord candidates, the following determination is carried out for five consecutive first chord candidates $M1(t-2)$, $M1(t-1)$, $M1(t)$, $M1(t+1)$, and $M1(t+2)$ and five second consecutive chord candidates $M2(t-2)$, $M2(t-1)$, $M2(t)$, $M2(t+1)$, and $M2(t+2)$ corresponding to the first candidates. More specifically, it is determined whether a relation represented by $M1(t-2)=M1(t+2)$, $M2(t-2)=M2(t+2)$, $M1(t-1)=M1(t)=M1(t+1)=M2(t-2)$, and $M2(t-1)=M2(t)=M2(t+1)=M1(t-2)$ is established. If the relation is established, $M1(t-1)=M1(t)=M1(t+1)=M1(t-2)$ and $M2(t-1)=M2(t)=M2(t+1)=M2(t-2)$ are determined, and the chords are exchanged between $M1(t-2)$ and $M2(t-2)$. Note that chords may be exchanged between $M1(t+2)$ and $M2(t+2)$ instead of between $M1(t-2)$ and $M2(t-2)$. It is also determined whether or not a relation represented by $M1(t-2)=M1(t+1)$, $M2(t-2)=M2(t+1)$, $M1(t-1)=M(t)=M1(t+1)=M2(t-2)$ and $M2(t-1)=M2(t)=M2(t+1)=M1(t-2)$ is established. If the relation is established, $M1(t-1)=M(t)=M1(t-2)$ and $M2(t-1)=M2(t)=M2(t-2)$ are determined, and the chords are exchanged between $M1(t-2)$ and $M2(t-2)$. The chords may be exchanged between $M1(t+1)$ and $M2(t+1)$ instead of between $M1(t-2)$ and $M2(t-2)$.

The first chord candidates $M1(0)$ to $M1(R)$ and the

second chord candidates $M2(0)$ to $M2(R)$ read out in step S41, for example, change with time as shown in Fig. 11, the averaging in step S42 is carried out to obtain a corrected result as shown in Fig. 12. In addition, the chord exchange in step S43 corrects the fluctuations of the first and second chord candidates as shown in Fig. 13. Note that Figs. 11 to 13 show changes in the chords by a line graph in which positions on the vertical line correspond to the kinds of chords.

The candidate $M1(t)$ at a chord transition point t of the first chord candidates $M1(0)$ to $M1(R)$ and $M2(t)$ at the chord transition point t of the second chord candidates $M2(0)$ to $M2(R)$ after the chord exchange in step S43 are detected (step S44), and the detection point t (4 bytes) and the chord (4 bytes) are stored for each of the first and second chord candidates in the data storing device 5 (step S45). Data for one music piece stored in step S45 is chord progression music data. These steps S41 to S45 correspond to a smoothing device (FOR EP: smoothing means).

When the first and second chord candidates $M1(0)$ to $M1(R)$ and $M2(0)$ to $M2(R)$, after exchanging the chords in step S43, fluctuate with time as shown in Fig. 14A, the time and chords at transition points are extracted as data. Fig. 14B shows the content of data at transition points among the first chord candidates F, G, D, Bb (B flat), and F that are expressed as hexadecimal data 0x08, 0x0A, 0x05, 0x01, and 0x08. The transition points t are $T1(0)$, $T1(1)$, $T1(2)$,

T1(3), and T1(4). Fig. 14C shows data contents at transition points among the second chord candidates C, Bb, F#m, Bb, and C that are expressed as hexadecimal data 0x03, 0x01, 0x29, 0x01, and 0x03. The transition points t are T2(0), T2(1), T2(2), T2(3), and T2(4). The data contents shown in Figs. 14B and 14C are stored together with the identification information of the music piece in the data storing device 5 in step S45 as a file in the form as shown in Fig. 14D.

The chord analysis operation described above is repeatedly carried out for audio signals representing sounds of different music pieces, so that chord progression music data is stored in the data storing device 5 as files for a plurality of music pieces. Note that music data of PCM signals corresponding to the chord progression music data in the data storing device 5 is stored in the data storing device 4.

A first chord candidate in a chord transition point among the first chord candidates and a second chord candidate in a chord transition point among second chord candidates are detected in step S44, and they are final chord progression music data. Therefore, the capacity per music piece can be reduced even as compared to compression data such as MP3-formatted data, and data for each music piece can be processed at high speed.

The chord progression music data written in the data storing device 5 is chord data temporally in synchronization

with the actual music. Therefore, when the chords are actually reproduced by the music reproducing device 10 using only the first chord candidate or the logical sum output of the first and second chord candidates, the accompaniment can be played to the music.

Now, the operation of detecting the structure of a music piece stored in the data storing device 5 as chord progression music data will be described. The music structure detection operation is carried out by the chord progression comparison device 7 and the repeating structure detection device 8.

As shown in Fig. 15, in the music structure detection operation, first chord candidates $M1(0)$ to $M1(a-1)$ and second chord candidates $M2(0)$ to $M2(b-1)$ for a music piece whose structure is to be detected are read out from the data storing device 5 serving as the storing means (step S51). The music piece whose structure is to be detected is, for example, designated by operating the input operation device 2. The letter a represents the total number of the first chord candidates, and b represents the total number of the second chord candidates. First chord candidates $M1(a)$ to $M1(a+K-1)$ and second chord candidates $M2(b)$ to $M2(b+K-1)$ each as many as K are provided as temporary data (step S52). Here, if $a < b$, the total chord numbers P of the first and second chord candidates in the temporary data are each equal to a , and if $a \geq b$, the total chord number P is equal to b . The temporary data is added following the first chord

candidates $M1(0)$ to $M1(a-1)$ and second chord candidates $M2(0)$ to $M2(b-1)$.

First chord differential values $MR1(0)$ to $MR1(P-2)$ are calculated for the read out first chord candidates $M1(0)$ to $M1(P-1)$ (step S53). The first chord differential values are calculated as $MR1(0)=M1(1)-M1(0)$, $MR1(1)=M1(2)-M1(1)$, ... , and $MR1(P-2)=M1(P-1)-M1(P-2)$. In the calculation, it is determined whether or not the first chord differential values $MR1(0)$ to $MR1(P-2)$ are each smaller than zero, and 12 is added to the first chord differential values that are smaller than zero. Chord attributes $MA1(0)$ to $MA1(P-2)$ after chord transition are added to the first chord differential values $MR1(0)$ to $MR1(P-2)$, respectively. Second chord differential values $MR2(0)$ to $MR2(P-2)$ are calculated for the read out second chord candidates $M2(0)$ to $M2(P-1)$ (step S54). The second chord differential values are calculated as $MR2(0)=M2(1)-M2(0)$, $MR2(1)=M2(2)-M2(1)$, ..., and $MR2(P-2)=M2(P-1)-M2(P-2)$. In the calculation, it is determined whether or not the second chord differential values $MR2(0)$ to $MR2(P-2)$ are each smaller than zero, and 12 is added to the second chord differential values that are smaller than zero. Chord attributes $MA2(0)$ to $MA2(P-2)$ after the chord transition are added to the second chord differential values $MR2(0)$ to $MR2(P-2)$, respectively. Note that values shown in Fig. 9B are used for the chord attributes $MA1(0)$ to $MA1(P-2)$, and $MA2(0)$ to $MA2(P-2)$.

Fig. 16 shows an example of the operation in steps S53

and S54. More specifically, when the chord candidates are in a row of Am7, Dm, C, F, Em, F, and Bb# (B flat sharp), the chord differential values are 5, 10, 5, 11, 1, and 5, and the chord attributes after transition are 0x02, 0x00, 0x00, 0x02, 0x00, and 0x00. Note that if the chord attribute after transition is 7th, major is used instead. This is for the purpose of reducing the amount of operation because the use of 7th hardly affects a result of the comparison operation.

After step S54, the counter value c is initialized to zero (step S55). Chord candidates (partial music data pieces) as many as K (for example 20) starting from the c -th candidate are extracted each from the first chord candidates $M1(0)$ to $M1(P-1)$ and the second chord candidates $M2(0)$ to $M2(P-1)$ (step S56). More specifically, the first chord candidates $M1(c)$ to $M1(c+K-1)$ and the second chord candidates $M2(c)$ to $M2(c+K-1)$ are extracted. Here, $M1(c)$ to $M1(c+K-1)=U1(0)$ to $U1(K-1)$, and $M2(c)$ to $M2(c+K-1)=U2(0)$ to $U2(K-1)$. Fig. 17 shows how $U1(0)$ to $U1(K-1)$ and $U2(0)$ to $U2(K-1)$ are related to the chord progression music data $M1(0)$ to $M1(P-1)$ and $M2(0)$ to $M2(P-1)$ to be processed and the added temporary data.

After step S56, first chord differential values $UR1(0)$ to $UR1(K-2)$ are calculated for the first chord candidates $U1(0)$ to $U1(K-1)$ for the partial music data piece (step S57). The first chord differential values in step S57 are calculated as $UR1(0)=U1(1)-U1(0)$, $UR1(1)=U1(2)-U1(1)$, ...,

and $UR1(K-2)=U1(K-1)-U1(K-2)$. In the calculation, it is determined whether or not the first chord differential values $UR1(0)$ to $UR1(K-2)$ are each smaller than zero, and 12 is added to the first chord differential values that are smaller than zero. Chord attributes $UA1(0)$ to $UA1(K-2)$ after the chord transition are added to the first chord differential values $UR1(0)$ to $UR1(K-2)$, respectively. The second chord differential values $UR2(0)$ to $UR2(K-2)$ are calculated for the second chord candidates $U2(0)$ to $U2(K-1)$ for the partial music data piece, respectively (step S58). The second chord differential values are calculated as $UR2(0)=U2(1)-U2(0)$, $UR2(1)=U2(2)-U2(1)$, ..., and $UR2(K-2)=U2(K-1)-U2(K-2)$. In the calculation, it is also determined whether or not the second chord differential values $UR2(0)$ to $UR2(K-2)$ are each smaller than zero, and 12 is added to the second chord differential values that are smaller than zero. Chord attributes $UA2(0)$ to $UA2(K-2)$ after chord transition are added to the second chord differential values $UR2(0)$ to $UR2(K-2)$, respectively.

Cross correlation operation is carried out based on the first chord differential values $MR1(0)$ to $MR1(K-2)$ and the chord attributes $MA1(0)$ to $MA1(K-2)$ obtained in the step S53, K first chord candidates $UR1(0)$ to $UR1(K-2)$ starting from the c-th candidate and the chord attributes $UA1(0)$ to $UA1(K-2)$ obtained in step S57, and K second chord candidates $UR2(0)$ to $UR2(K-2)$ starting from the c-th candidate and the chord attributes $UA2(0)$ to $UA2(K-2)$ obtained in step S58

(step S59). In the cross correlation operation, the correlation coefficient $COR(t)$ is produced from the following expression (3). The smaller the correlation coefficient $COR(t)$ is, the higher the similarity is.

$$\begin{aligned}
 COR(t) = & \sum 10(|MR1(t+k)-UR1(k')|+|MA1(t+k)-UA1(k')| \\
 & +|WM1(t+k+1)/WM1(t+k)-WU1(k'+1)/WU1(k')|) \\
 & +\sum 10(|MR1(t+k)-UR2(k')|+|MA1(t+k)-UA2(k')| \\
 & +|WM1(t+k+1)/WM1(t+k)-WU2(k'+1)/WU2(k')|) \\
 & \dots(3)
 \end{aligned}$$

where $WU1()$, $WM1()$, and $WU2()$ are time widths for which the chords are maintained, $t = 0$ to $P-1$, and \sum operations are for $k = 0$ to $K-2$ and $k' = 0$ to $K-2$.

The correlation coefficient $COR(t)$ in step S59 is produced as t is in the range from 0 to $P-1$. In the operation of the correlation coefficient $COR(t)$ in step S59, a jump process is carried out. In the jump process, the minimum value for $MR1(t+k+k1)-UR1(k'+k2)$ or $MR1(t+k+k1)-UR2(k'+k2)$ is detected. The values $k1$ and $k2$ are each an integer in the range from 0 to 2. More specifically, as $k1$ and $k2$ are changed in the range from 0 to 2, the point where $MR1(t+k+k1)-UR1(k'+k2)$ or $MR1(t+k+k1)-UR2(k'+k2)$ is minimized is detected. The value $k+k1$ at the point is set as a new k , and $k'+k2$ is set as a new k' . Then, the correlation coefficient $COR(t)$ is calculated according to the expression (3).

If chords after respective chord transitions at the same point in both of the chord progression music data to be

processed and K partial music data pieces from the c-th piece of the chord progression music data are either C or Am or either Cm or Eb (E flat), the chords are regarded as being the same. More specifically, as long as the chords after the transitions is chords of a related key, $|MR1(t+k)-UR1(k')|+|MA1(t+k)-UA1(k')|=0$ or $|MR1(t+k)-UR2(k')|+|MA1(t+k)-UA2(k')|=0$ in the above expression stands. For example, the transform of data from chord F to major by a difference of seven degrees, and the transform of the other data to minor by a difference of four degrees are regarded as the same. Similarly, the transform of data from chord F to minor by a difference of seven degrees and the transform of the other data to major by a difference of ten degrees are treated as the same.

The cross-correlation operation is carried out based on the second chord differential values $MR2(0)$ to $MR2(K-2)$ and the chord attributes $MA2(0)$ to $MA2(K-2)$ obtained in step S54, and K first chord candidates $UR1(0)$ to $UR1(K-2)$ from c-th candidate and the chord attributes $UA1(0)$ to $UA1(K-2)$ obtained in step S57, and K second chord candidates $UR2(0)$ to $UR2(K-2)$ from the c-th candidate and the chord attributes $UA2(0)$ to $UA2(K-2)$ obtained in step S58 (step S60). In the cross-correlation operation, the correlation coefficient $COR'(t)$ is calculated by the following expression (4). The smaller the correlation coefficient $COR'(t)$ is, the higher the similarity is.

$$COR'(t) = \sum 10(|MR2(t+k)-UR1(k')|+|MA2(t+k)-UA1(k')|$$

$$\begin{aligned}
& + |WM2(t+k+1)/WM2(t+k) - WU1(k'+1)/WU1(k')|) \\
& + \sum 10(|MR2(t+k) - UR2(k')| + |MA2(t+k) - UA2(k')| \\
& + |WM2(t+k+1)/WM2(t+k) - WU2(k'+1)/WU2(k')|) \\
& \dots (4)
\end{aligned}$$

where $WU1()$, $WM2()$, and $WU2()$ are time widths for which the chords are maintained, $t = 0$ to $P-1$, \sum operations are for $k = 0$ to $K-2$ and $k' = 0$ to $K-2$.

The correlation coefficient $COR'(t)$ in step S60 is produced as t changes in the range from 0 to $P-1$. In the operation of the correlation coefficient $COR(t)$ in step S60, a jump process is carried out similarly to step S59 described above. In the jump process, the minimum value for $MR2(t+k+k1) - UR1(k'+k2)$ or $MR2(t+k+k1) - UR2(k'+k2)$ is detected. The values $k1$ and $k2$ are each an integer from 0 to 2. More specifically, $k1$ and $k2$ are each changed in the range from 0 to 2, and the point where $MR2(t+k+k1) - UR1(k'+k2)$ or $MR2(t+k+k1) - UR2(k'+k2)$ is minimized is detected. Then, $k+k1$ at the point is set as a new k , and $k'+k2$ is set as a new k' . Then, the correlation coefficient $COR'(t)$ is calculated according to the expression (4).

If chords after respective chord transitions at the same point in both of the chord progression music data to be processed and the partial music data piece are either C or Am or either Cm or Eb, the chords are regarded as being the same. More specifically, as long as the chords after the transitions are chords of a related key, $|MR2(t+k) - UR1(k')| + |MA2(t+k) - UA1(k')| = 0$ or $|MR2(t+k) - UR2(k')| + |MA2(t+k) - UA2(k')|$

$l=0$ in the above expression stands.

Fig. 18A shows the relation between chord progression music data to be processed and its partial music data pieces. In the partial music data pieces, the part to be compared to the chord progression music data changes as t advances. Fig. 18B shows changes in the correlation coefficient $COR(t)$ or $COR'(t)$. The similarity is high at peaks in the waveform.

Fig. 18C shows time widths $WU(1)$ to $WU(5)$ during which the chords are maintained, a jump process portion and a related key portion in a cross-correlation operation between the chord progression music data to be processed and its partial music data pieces. The double arrowhead lines between the chord progression music data and partial music data pieces point at the same chords. The chords connected by the inclined arrow lines among them and not present in the same time period represent chords detected by the jump process. The double arrowhead broken lines point at chords of related keys.

The cross-correlation coefficients $COR(t)$ and $COR'(t)$ calculated in steps S59 and S60 are added to produce a total cross correlation coefficient $COR(c, t)$ (step S61). More specifically, $COR(c, t)$ is calculated by the following expression (5):

$$COR(c, t) = COR(t) + COR'(t) \text{ where } t=0 \text{ to } P-1 \quad \dots (5)$$

Figs. 19A to 19F each show the relation between phrases (chord progression row) in a music piece represented by

chord progression music data to be processed, a phrase represented by a partial music data piece, and the total correlation coefficient $COR(c, t)$. The phrases in the music piece represented by the chord progression music data are arranged like A, B, C, A', C', D, and C" in the order of the flow of how the music goes after introduction I that is not shown. The phrases A and A' are the same and the phrases C, C', and C" are the same. In Fig. 19A, phrase A is positioned at the beginning of the partial music data piece, and $COR(c, t)$ generates peak values indicated with \square in the points corresponding to phrases A and A' in the chord progression music data. In Fig. 19B, phrase B is positioned at the beginning of the partial music data piece, and $COR(c, t)$ generates a peak value indicated with X in the point corresponding to phrase B in the chord progression music data. In Fig. 19C, phrase C is positioned at the beginning of the partial music data piece, and $COR(c, t)$ generates peak values indicated with O in the points corresponding to phrases C, C', and C" in the chord progression music data. In Fig. 19D, phrase A' is positioned at the beginning of the partial music data piece, and $COR(c, t)$ generates peak values indicated with \square in points corresponding to phrases A and A' in the chord progression music data. In Fig. 19E, phrase C' is positioned at the beginning of the partial music data piece, and $COR(c, t)$ generates peak values indicated with O in the points corresponding to phrases C, C' and C" in the chord progression music data. In Fig. 19F,

phrase C" is positioned at the beginning of the partial music data piece, and $COR(c, t)$ generates peak values indicated with 0 in the points corresponding to phrases C, C', and C" in the chord progression music data.

After step S61, the counter value c is incremented by one (step S62), and it is determined whether or not the counter value c is greater than $P-1$ (step S63). If $c \leq P-1$, the correlation coefficient $COR(c, t)$ has not been calculated for the entire chord progression music data to be processed. Therefore, the control returns to step S56 and the operation in steps S56 to S63 described above is repeated.

If $c > P-1$, $COR(c, t)$, i.e., the peak values for $COR(0, 0)$ to $COR(P-1, P-1)$ are detected, and $COR_PEAK(c, t)=1$ is set for c and t when the peak value is detected, while $COR_PEAK(c, t)=0$ is set for c and t when the value is not a peak value (step S64). The highest value in the part above a predetermined value for $COR(c, t)$ is the peak value. By the operation in step S64, the row of $COR_PEAK(c, t)$ is formed. Then in the $COR_PEAK(c, t)$ row, the total value of values for $COR_PEAK(c, t)$ as t changes from 0 to $P-1$ is calculated as the peak number $PK(t)$ (step S65).

$PK(0)=COR_PEAK(0, 0)+COR_PEAK(1, 0)+\dots COR_PEAK(P-1, 0)$,
 $PK(1)=COR_PEAK(0, 1)+COR_PEAK(1, 1)+\dots COR_PEAK(P-1, 1)$,
 \dots , $PK(P-1)=COR_PEAK(0, P-1)+COR_PEAK(1, P-1)+\dots COR_PEAK(P-1, P-1)$. Among peak numbers $PK(0)$ to $PK(P-1)$, at least two consecutive identical number ranges are separated as

identical phrase ranges, and music structure data is stored in the data storing device 5 accordingly (step S66). If for example the peak number $PK(t)$ is two, it means the phrase is repeated twice in the music piece, and if the peak number $PK(t)$ is three, the phrase is repeated three times in the music piece. The peak numbers $PK(t)$ within an identical phrase range are the same. If the peak number $PK(t)$ is one, the phrase is not repeated.

Fig. 20 shows peak numbers $PK(t)$ for a music piece having phrases I, A, B, C, A', C', D, and C" shown in Figs. 19A to 19F and positions $COR_PEAK(c, t)$ where peak values are obtained on the basis of the calculation result of the cross correlated coefficient $COR(c, t)$. $COR_PEAK(c, t)$ is represented in a matrix, the abscissa represents the number of chords $t=0$ to $P-1$, and the ordinate represents the starting positions $c=0$ to $P-1$ for partial music data pieces. The dotted part represents the position corresponding to $COR_PEAK(c, t)=1$ where $COR(c, t)$ attains a peak value. A diagonal line represents self correlation between the same data, and therefore shown with a line of dots. A dot line in the part other than the diagonal lines corresponds to phrases according to repeated chord progression. With reference to Figs. 19A to 19F, X corresponds to phrases I, B, and D that are performed only once, O represents three-time repeating phrases C, C', and C", and \square corresponds to twice-repeating phrases A and A'. The peak number $PK(t)$ is 1, 2, 1, 3, 2, 3, 1, and 3 for phrases I, A, B, C, A', C',

D, and C", respectively. This represents the music piece structure as a result.

The music structure data has a format as shown in Fig. 21. Chord progression music data $T(t)$ shown in Fig. 14C is used for the starting time and ending time information for each phrase.

The music structure detection result is displayed at the display device 9 (step 67). The music structure detection result is displayed as shown in Fig. 22, so that each repeating phrase part in the music piece can be selected. Music data for the repeating phrase part selected using the display screen or the most frequently repeating phrase part is read out from the music data storing device 4 and supplied to the music reproducing device 10 (step S68). In this way, the music reproducing device 10 sequentially reproduces the supplied music data, and the reproduced data is supplied to the digital-analog converter 11 as a digital signal. The signal is converted into an analog audio signal by the digital-analog converter 11 and then reproduced sound of the repeating phrase part is output from the speaker 12.

Consequently, the user can be informed of the structure of the music piece from the display screen and can easily listen to a selected repeating phrase or the most frequently repeating phrase in the music piece of the process object.

Step S56 in the above music structure detection operation corresponds to the partial music data producing device (FOR EP: partial music data producing means). Steps

S57 to S63 correspond to the comparison means for calculating similarities (cross correlation coefficient $COR(c, t)$), step S64 corresponds to the chord position detector (FOR EP: chord position detection means), and steps S65 to S68 correspond to the output device (FOR EP: output means).

The jump process and related key process described above are carried out to eliminate the effect of extraneous noises or the frequency characteristic of an input device when chord progression music data to be processed is produced on the basis of an analog signal during the operation of the differential value before and after the chord transition. When rhythms and melodies are different between the first and second parts of the lyrics or there is a modulated part even for the same phrase, data pieces do not completely match in the position of chords and their attributes. Therefore, the jump process and related key process are also carried out to remedy the situation. More specifically, if the chord progression is temporarily different, similarities can be detected in the tendency of chord progression within a predetermined time width, and therefore it can accurately be determined whether the music data belongs to the same phrase even when the data pieces have different rhythms or melodies or have been modulated. Furthermore, by the jump process and related key process, accurate similarities can be obtained in cross-correlation operations for the part other than the part subjected to

these processes.

Note that in the above embodiment, the invention is applied to music data in the PCM data form, but when a row of notes included in a music piece are known in the processing in step S28, MIDI data may be used as the music data. Furthermore, the system according to the embodiment described above is applicable in order to sequentially reproduce only the phrase parts repeating many times in the music piece. In other words, a highlight reproducing system for example can readily be implemented.

Fig. 23 shows another embodiment of the invention. In the music processing system in Fig. 23, the chord analysis device 3, the temporary memory 6, the chord progression comparison device 7 and the repeating structure detection device 8 in the system in Fig. 1 are formed by the computer 21. The computer 21 carries out the above chord analysis operation and the music structure detection operation in response to a program stored in the storing device 22. The storing device 22 does not have to be a hard disk drive and may be a drive for a storage medium. In the case, chord progression music data may be written in the storage medium.

As in the foregoing, according to the invention, the structure of a music piece including repeating parts can appropriately be detected with a simple structure.

This application is based on a Japanese Patent Application No. 2002-352865 which is hereby incorporated by reference.